Generalized RBF Neural Network and FEM for Material Characterization Through Inverse Analysis

Tarik Hacib, Mohammed Rachid Mekideche, Fouzia Moussouni, Nassira Ferkha, and Stéphane Brisset

Abstract This paper describes a new methodology for using artificial neural networks (ANN) and finite element method (FEM) in an electromagnetic inverse problem (IP) of parameters identification. The approach is used to identify unknown parameters of ferromagnetic materials. The methodology used in this study consists in the simulation of a large number of parameters in a material under test, using the FEM. Both variations in relative magnetic permeability and electric conductivity of the material under test are considered. Then, the obtained results are used to generate a set of vectors for the training of generalized radial basis function neural networks (RBFNN). Finally, the obtained neural network (NN) is used to evaluate a group of new materials, simulated by the FEM, but not belonging to the original dataset. The reached results demonstrate the efficiency of the proposed approach.

1 Introduction

IPs in electromagnetic are usually formulated and solved as optimization problems, so iterative methods are commonly used approaches to solve this kind of problems [1]. These methods involve solving well behaved forward problem in a feedback loop. The numerical models such as FE model are used to represent the forward process. However, iterative methods using the numerical based forward models are computationally expensive. Parameters identification using NNs can be recast as a problem in multidimensional interpolation, which consists of finding the unknown nonlinear relationship between inputs and outputs in a space spanned

T. Hacib, M. R. Mekideche, and N. Ferkha
Laboratoire L.A.M.E.L, Université de Jijel, BP 98 Ouled Aissa Jijel, Algeria
tarik_hacib@mail.univ-jijel.dz

F. Moussouni and S. Brisset
Laboratoire L2EP – Ecole Centrale de Lille, Cité Scientifique – BP 48
F59651 Villeneuve d'Ascq cedex, France

T. Hacib et al.: *Generalized RBF Neural Network and FEM for Material Characterization Through Inverse Analysis*, Studies in Computational Intelligence (SCI) **119**, 11–19 (2008)
www.springerlink.com

by the activation functions associated with the NN nodes [2, 3]. The input space corresponds to the signal generated by sensors and the output corresponds to the electromagnetic parameters such as relative magnetic permeability and electrical conductivity.

In this paper, we propose a new method for the robust identification of electromagnetic properties. The method is based on the FEM and a RBFNN scheme. We present a comparison of the results obtained using the proposed method with those obtained from a multilayer perceptron (MLP). It is shown that the generalized RBFNN is faster both in training as well as identification of parameters.

2 Neural Networks

NNs are connectionist models proposed in an attempt to mimic the function of the human brain. A NN consists of a large number of simple processing elements called neurons [1]. Neurons implement simple functions on their inputs and are massively interconnected by means of weighted interconnections. These weights, estimated by means of a training process, determine the functionality of the NN. The training process uses a training database to determine the network parameters (weights). NNs have been widely used for function approximation and multidimensional interpolation [4]. Given a set of p ordered pairs $(x_i,\ d_i)\,, i = 1, 2, \ldots, p$ with $x_i \in R^N$ and $d_i \in R$, the problem of interpolation is to find a function $F : R^N \rightarrow R^1$ that satisfies the interpolation condition

$$F\,(x_i) = d_i,\ i = 1, \ldots, p. \tag{1}$$

For strict interpolation, the function F is constrained to pass through all the p data points. The definition can be easily extended to the case where the output is M-dimensional. The desired function is then $F : R^N \rightarrow R^M$. In practice, the function F is unknown and must be determined from the given data $(x_i, d_i)\,, i = 1, 2, \ldots,\ p$. A typical NN implementation of this problem is a two step process: training, where the NN learns the function F given the training data $\{x_i, d_i\}$, and generalization, where the NN predicts the output for a test input. Networks using two different types of basis functions (BFs) are described in the following sections.

2.1 Radial Basis Function Neural Networks

An RBFNN consists of an input and output layer of nodes and a single hidden layer (Fig. 1). Each node in the hidden layer implements a BF $G\,(x, x_i)$ and the number of hidden nodes is equal to the number of data points in the training database. The RBFNN approximates the unknown function that maps the input to the output in terms of a BF expansion, with the functions $G\,(x, x_i)$ as the BFs. The I/O relation for RBFNN is given by

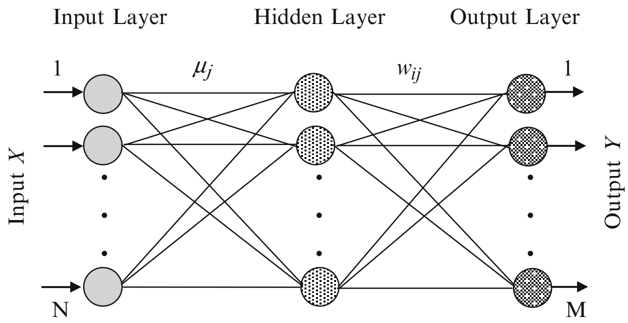

Fig. 1 Three layer feed-forward network

$$y_l = \sum_{j=1}^{N} w_{lj} G(x, x_j),\ l = 1, 2 \ldots, M. \tag{2}$$

Where N is the number of BFs used, $\mathrm{y} = (y_1,\ y_2, \ldots,\ y_M)^T$ is the output of the RBFNN, x is the test input, x_j is the center of the BF, and w_{lj} are the expansion coefficients or weights. Each training data sample is selected as the center of a BF. BFs $G(x, x_i)$ that are radially symmetric are called radial BFs. Commonly used radial BFs include the Gaussian and inverse multiquadrics [2].

The network described above is called an exact RBFNN, since each training data point is used as a basis center. The storage costs of an exact RBFNN can be enormous, especially when the training database is large. An alternative to an exact RBFNN is a generalized RBFNN, where the number of BFs is less than the number of training data points. The problem then changes from strict interpolation (in an exact RBFNN) to an approximation, where certain error constraints are to be satisfied. The operation of the generalized RBFNN is summarized in the following steps.

2.2 Center Selection

This is achieved by using either the k-means clustering algorithm [5] or other optimization techniques that select the BF locations by minimizing the error in the approximation. The I/O relation for a generalized RBFNN using Gaussian BFs is given by

$$y_l = \sum_{j=1}^{H} w_{lj} \exp\left(-\frac{\|x - c_j\|^2}{2\sigma_j^2}\right), \tag{3}$$

where H is the total number of BFs used, c_j is the center of the Gaussian BF, and σ_j is the width of the Gaussian. The NN architecture is then selected by setting the number of input nodes equal to the input dimension, the number of hidden nodes to the number of centers obtained in this step, and the number of output nodes equal to the output dimension.

2.3 Training of Generalised RBF Neural Network

Training of the NN involves determining the weights w_{lj}, in addition to the centers and widths of the BFs. Writing (2) in matrix-vector form as

$$\mathbf{Y} = \mathbf{GW}, \tag{4}$$

Equation (4) can be solved for $\mathbf{W}$ as:

$$\mathbf{W} = \mathbf{G}^{+}\mathbf{Y}, \tag{5}$$

where $\mathbf{G}^{+}$ is the pseudo-inverse.

2.4 Generalization

In the test phase, the unknown pattern x is mapped using the relation

$$F(x) = \sum_{j=1}^{H} w_{lj}\, exp\left(-\frac{\left\|x - c_j\right\|^2}{2\sigma_j^2}\right). \tag{6}$$

3 Electromagnetic Field Computation

In this study, the magnetic field is calculated using the FEM. This method is based on the A representation of the magnetic field. The calculations are performed in two steps. First, the magnetic field intensity is calculated by solving the equation:

$$rot\left(\frac{1}{\mu} rot(A)\right) + \mathrm{j}\omega\sigma A = J, \tag{7}$$

where μ is the magnetic permeability, σ the electric conductivity and $\boldsymbol{J}$ the electric current density.

Equation (7) is discretized using the Galerkin FEM, which leads to the following algebraic matrix equation:

$$([\boldsymbol{K}] + \mathrm{j}\omega[\boldsymbol{C}])[\boldsymbol{A}] = [\boldsymbol{F}]. \tag{8}$$

In the second step, the field solution is used to calculate the magnetic induction $\boldsymbol{B}$. More details about the FE theory can be found in [6].

4 Methodology for Parameters Identification

First of all, an electromagnetic device was idealized to be used as an electromagnetic field exciter (Fig. 2). In this paper, we have considered direct current in the coils. To increase the sensitivity of the electromagnetic device a magnetic core with a high permeability is used and the air gap between the core and the metallic wall is reduced to a minimum. Deviations of the magnetic induction (difference in magnetic induction without and with material under test) at equally stepped points in the external surface of the material under test are taken.

Figure 3 show the steps of the methodology used in this work. Steps 1–4 correspond to the FE analysis.

The simulations were done for a hypothetic metallic wall with 1 mm height and 15 mm width. The material of the metallic wall is 1006 Steel (a magnetic material). The relative magnetic permeability of the core is supposed to be 2,500 and the air gap is 0.1 mm.

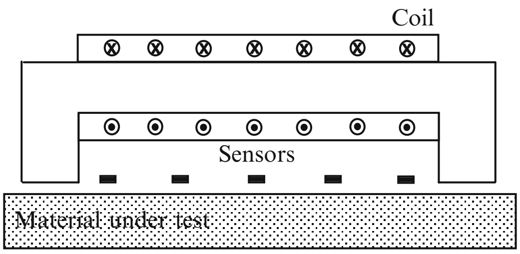

Fig. 2 Arrangement for the measurements

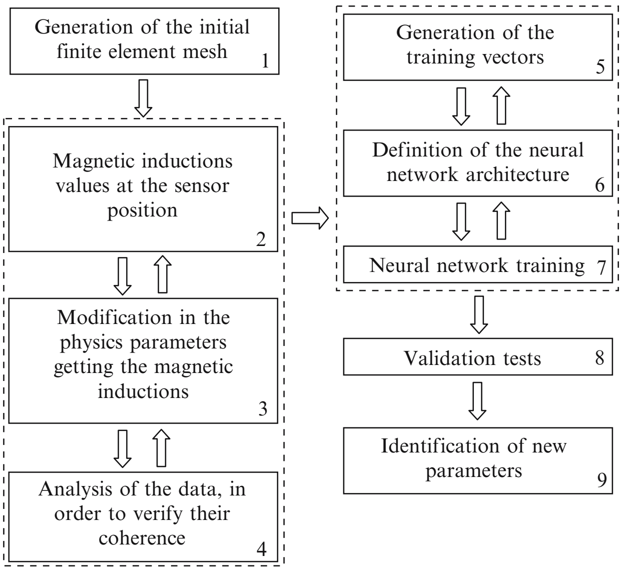

Fig. 3 Flowchart of the used methodology

During the phase of FEs simulations, errors can appear, due to it's massively nature. So, the results of the simulations must be carefully analyzed. This can be done, for instance, plotting in the same graphic the magnetic induction deviations for a set of parameters. Figure 4 shows the magnetic induction deviation at the sensor position for three materials having the same electrical conductivity (10^3 [S m^{-1}]), and relative magnetic permeability ranging from 50 to 300. Figure 5 shows the graphics for a fixed magnetic relative permeability (240), and three different electrical conductivity ranging from 6×10^5 [S m^{-1}] to 10^8 [S m^{-1}].

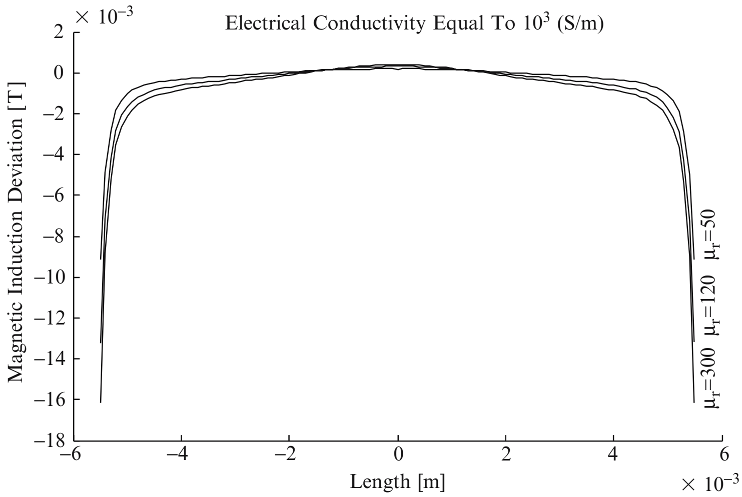

Fig. 4 Magnetic induction deviation for three values of magnetic relative. Permeability and electrical conductivity equal to 10^3 (S m^{-1})

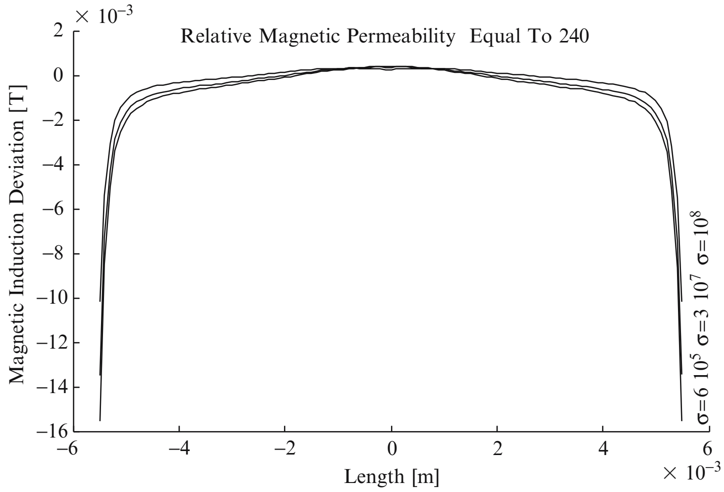

Fig. 5 Magnetic induction deviation for three values of electrical conductivity and magnetic relative permeability equal to 240

5 Formulation of Network Models for Parameters Identifications

In the step 5, we generate the training vectors for NNs. In this work, we generated 300 vectors for NNs training. Each of the vectors consists of 11 input values, which represent the deviation of magnetic induction, and two output values, which represent the relative magnetic permeability and electrical conductivity. Of the 300 vectors, a random sample of 225 cases (75%) was used as training, 75 (25%) for validation.

To show stability of the proposed approach, the measured values, which intrinsically contains errors in the real word, is obtained by adding a random perturbation to the exact inputs values

$$\tilde{In} = In_{exact} + \delta\lambda, \tag{9}$$

where δ is the standard deviation and λ is a random variable taken from a Gaussian distribution.

MLP network architecture considered for this application was a single hidden layer with sigmoid activation function. A back-propagation algorithm based on Levenberg–Marquardt optimization technique [7] was used to train the MLP network for the above data. The MLP architecture had 11 input variables, one hidden layer with 24 hidden nodes and two output nodes. Total number of weights present in the model was 338. The best MLP was obtained at lowest mean square error (MSE) of 10^{-6}. Percentage correct prediction of the MLP model was 95.6%. Generalized RBFNN performed best at 140 centres and Gaussian BFs. MSE using the best centres and Gaussian BFs was 5×10^{-6}. Percentage correct prediction of the generalized RBFNN model was 98.2%. Figure 6 shows the performance of the generalized RBFNN during a training session. Table 1 shows some results for the validation of the network by using 5% of noise ($\delta = 0.05$), for this session.

As we can see, the results obtained in the validation are very close to the expected ones. The worse identification parameter was obtained with MLP network.

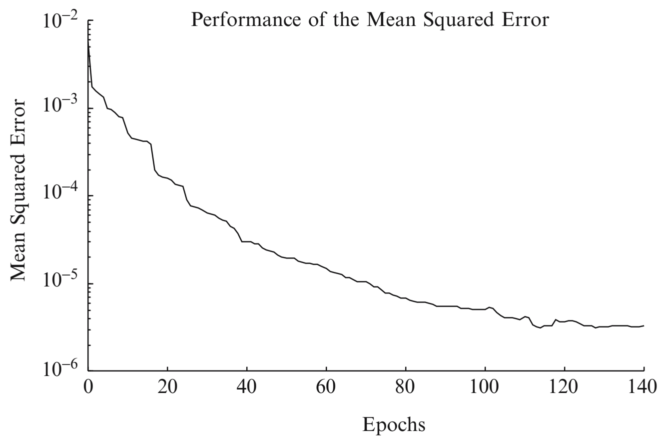

Fig. 6 Performance of the generalized RBFNN during a training session

Table 1 Expected and obtained values during a training session

Relative magnetic permeability, μ_r			Electric conductivity, σ (S m^{-1})		
Expected	Generalized RBF	MLP	Expected	Generalized RBF	MLP
83.34	83.329	83.321	7.020 e006	7.018 e006	7.017 e006
287.55	287.54	287.53	17.99 e003	18.01 e003	18.02 e003
311.34	311.35	311.35	4.500 e002	4.506 e002	4.519 e002
446.21	446.21	446.24	22.50 e006	22.49 e006	22.46 e006
527.07	527.04	527.33	3.010 e005	3.011 e005	3.014 e005

Table 2 Simulation results, for new parameters

Material no.	Relative magnetic permeability, μ_r			Electric conductivity, σ (S m^{-1})		
	Expected	Obtained		Expected	Obtained	
		Generalized RBF	MLP		Generalized RBF	MLP
1	89	89.001	88.933	65.00	65.02	65.03
2	212	211.99	211.98	2.500 e002	2.501 e002	2.502 e002
3	360	360.02	359.95	2.200 e006	2.199 e006	2.201 e006
4	472	472.01	472.03	4.100 e005	4.099 e005	4.101 e005

6 New Parameter Identification

After the NNs training and respective validations, new electromagnetic parameters were simulated by the FEM, for posteriori identification by the networks. Table 2 shows the parameters values of material under test, and the obtained values, by the NNs.

As we can see, the results obtained in the identification of new parameters, obtained by the NNs agree very well with the expected ones.

7 Conclusion

In this paper we presented an investigation on the use of the FEM and Generalized RBFNN for the identification of metallic walls parameters, present in industrial plants. The obtained weights of the network can be embedded in electronic devices in order to identify electromagnetic parameters of real metallic walls. The proposed approach was found to be highly effective in identification of parameters in electromagnetic devices. Comparison of the result indicates that Generalized RBF is trained and identifies the electromagnetic parameters faster than MLP.

References

1. S. R. H. Hoole, Artificial neural networks in the solution of inverse electromagnetic field problems, *IEEE Trans. Magn.*, 29(2), pp. 1931–1934, 1993.
2. S. Haykin, Neural networks: a comprehensive foundation, Englewood Cliffs, NJ: Prentice-Hall, 1999.
3. A. Fanni and A. Montisci, A neural inverse problem approach for optimal design, *IEEE Trans. Magn.*, 39(3), pp. 1305–1308, 2003.
4. P. Ramuhalli, L. Udpa and S.S. Udpa, Finite element neural networks for solving differential equations, *IEEE Trans. Neural Netw.*, 16(6), pp. 1381–1392, 2005.
5. J. A. Hartigan, Clustering Algorithms, John Wiley and Sons, Agnetism, CA: Academic, 2000.
6. P. P. Silvester and R. L. Ferrari, Finite Elements for Electrical Engineers. Cambridge, University Press, 1996.
7. M. T. Hagan and M. Menhaj, Training feed-forward networks with the Levenberg–Marquardt algorithm, *IEEE Trans. Neural Netw.*, 5(6), pp. 989–993, 1994.

http://www.springer.com/978-3-540-78489-0

Intelligent Computer Techniques in Applied Electromagnetics
Wiak, S.; Krawczyk, A.; Dolezel, I. (Eds.)
2008, X, 304 p. 201 illus., Hardcover
ISBN: 978-3-540-78489-0